(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 August 2003 (07.08.2003)

PCT

(10) International Publication Number
WO 03/065234 A1

(51) International Patent Classification[7]: G06F 13/28

(21) International Application Number: PCT/EP01/15347

(22) International Filing Date:
27 December 2001 (27.12.2001)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **BEALE, John** [GB/GB]; 21 Recreation Road, Tilehurst, Reading, Berkshire RG30 4UB (GB).

(74) Agent: **UNGERER, Olaf**; Eisenführ, Speiser & Partner, Arnulfstr. 25, 80335 Munich (DE).

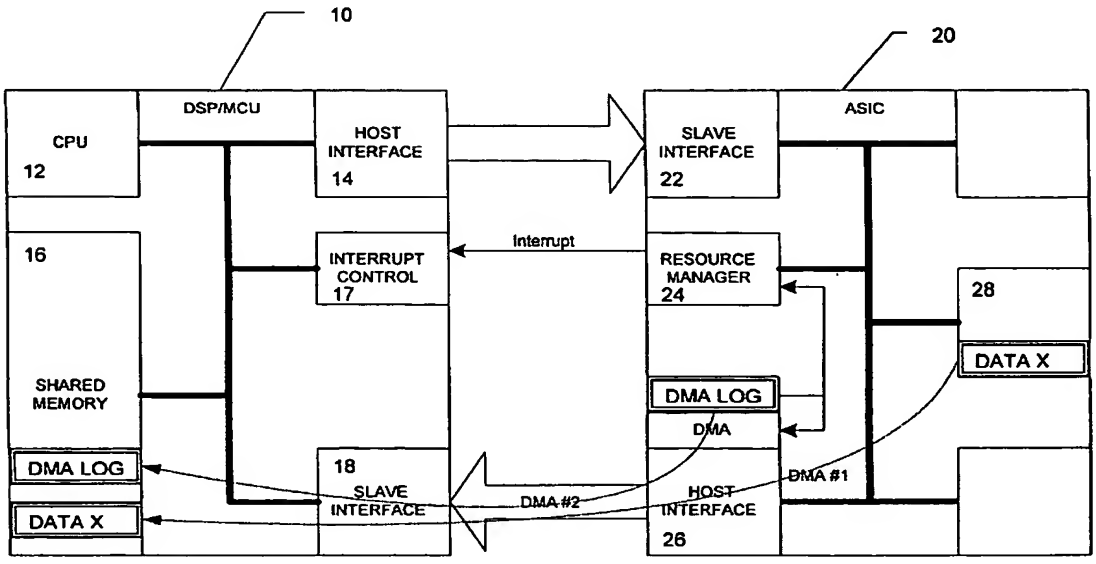
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: LOW-OVERHEAD PROCESSOR INTERFACING



(57) Abstract: The present invention relates to a method and system for performing a data trans-fer between a shared memory (16) of a processor device (10) and a circuitry (20) connected to the processor device (10), wherein the data transfer is performed by triggering a DMA transfer of the data to the processor device, adding the DMA transfer to a transaction log, and providing the transaction log to the processor device, when the transaction log has reached a predetermined depth limit. The processor device is then informed of the DMA transfer of the transaction log, so as to be able to validate the transferred data. Thereby, significant background data movement can be provided without introducing high core overheads at the proces-sor device (10).

WO 03/065234 A1

Low-Overhead Processor Interfacing

FIELD OF THE INVENTION

The present invention relates to a method and system for performing a data transfer between a memory of a processor device, such as a digital signal processor (DSP), and a circuitry, such as an application specific integrated circuit (ASIC), connected to the processor device.

BACKGROUND OF THE INVENTION

Distributed memory architectures are very good at implementing data flow processing, and data flow processing is in turn what almost every DSP and related application boils down to. Ideal fit between DSP and data flow systems is further strengthened by the need for DSP application to run in real-time, i.e. they must process data with a specified throughput and/or latency requirement. Shared memory systems may have difficulties in guaranteeing latency due to the uncertainties over memory contention.

If the central processing unit (CPU) of a DSP has to stop its current task and move data on and off the chip, performance will be poor. Therefore, direct memory access (DMA) controllers are provided for executing command sequences, auto-initialization and the like. In real-time data processing systems, this allows the DMA to run independently of the CPU. DMA requires primarily that the DSP does not access the memory involved. To achieve this, the CPU may be stopped or decoupled from the bus system to assure that the CPU and the DMA controller are not attempting to access the memories concurrently. During the DMA operation, the addresses fed to the memories are those generated by the DMA controller. After the DMA operation is completed, the addresses generated by the CPU once again determine which memory word is being accessed. Thus, DMA provides a data transfer which allows data to be moved between a peripheral controller and a system memory without interaction of the host CPU. The data may be moved by the peripheral controller itself, or by a separate third party DMA controller.

The stopping or decoupling of the CPU is usually performed based on interrupt routines triggered by external circuitries which intend to access the shared memory of the DSP. Thus, high interrupt overheads and associated DSP core load are associated with frequent interrupt service routines (ISRs) triggered in case of data

movements between the shared memory and external devices or circuitries connected to the DSP.

Document EP 0 908 830 A1 discloses a DSP-based communications adapter including a number of digital signal processors and network interface circuits for providing an attachment of a multi-channel telephone line. Each digital signal processor interrupts its host processor by transmitting an interrupt control block as data to a data memory of the host processor, and by subsequently sending an interrupt causing the host processor to examine the data memory. Thereby, a number of interrupts to the host processor from a single DSP is bundled and can be handled together. The overhead for individually handling each interrupt can thus be reduced. The interrupt blocks are written by means of a DMA operation to the memory of the host processor.

However, if this prior art solution is used for bundling interrupts of data movements between a shared memory of a DSP and an external circuitry, the CPU of the DSP still has to handle each interrupt of the interrupt block in order to trigger the corresponding ISRs required for data movement. Hence, overhead would still be a problem.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method and system for transferring data between a memory of a processor device and a circuitry connected to the processor device, by means of which interrupt overheads and associated core load can be reduced.

This object is achieved by a method of performing a data transfer between a memory of a processor device and a circuitry connected to the processor device, said method comprising the steps of:
setting up a direct memory access (DMA) for transferring data at said circuitry;
triggering a DMA transfer of said data to said processor device;
adding said DMA transfer to a transaction log;
providing said transaction log to said processor device when said transaction log has reached a predetermined depth limit; and
informing said processor device of the availability of said transaction log.

Furthermore, the above object is achieved by a processor device having a memory which can be accessed by a connected circuitry, said processor device being arranged to validate data, transferred to said memory by a direct memory access, based on a transaction log provided to said processor device.

Additionally, the above object is achieved by an integrated circuit having means for providing access to a processor device, said integrated circuit being arranged to set up a direct memory access for transferring data via said access means, to trigger a DMA transfer of said data, to add said DMA transfer to said transaction log, to provide said transaction log to said processor device when said transaction log has reached a predetermined depth limit, and to issue an information indicating the availability of said transaction log.

Moreover, the above object is achieved by a system for performing a data transfer between a memory of a processor device and a circuitry connected to said processor device,
wherein said circuitry is arranged to set up a direct memory access for transferring data, to trigger a DMA transfer of said data to said processor device, to add said DMA transfer to said transaction log, to provide said transaction log to said processor device when said transaction log has reached a predetermined depth limit, and to inform said processor device of the availability of said transaction log; and
wherein said processor device is arranged to validate said transferred data based on said available transaction log.

Accordingly, a pre-programmed DMA located on interfaces hosted by the external circuitry is provided to transfer data between the memory and the external circuitry, wherein processor control requirements can be reduced by using the transaction log. Thus, a plurality of data transfers can be bundled with a single DMA operation, since the processor device may validate or qualify the transferred data based on the available transaction log, when the availability, e.g. transfer or interrogation, of the transaction log has been informed to the processor device. Since the processor device is only involved in the signaling of the information regarding the provision, e.g. transfer or interrogation, of the transaction log, interrupt overheads and associated core load can be significantly reduced. Furthermore, the need for manual data movement is prevented, and a data rate matching can be provided between the shared memory and the on-chip bus system of the external circuitry to thereby reduced stalling of the system.

Due to the fact that multiple data structures held in the memory can be validated using a single processor involvement, e.g. by an interrupt service routine, core processing overhead is reduced especially when context switching is required under a real-time operating system (RTOS). Furthermore, since the transaction log can be stored locally within the memory of the processor device, slow core polling of the transaction status within the external circuitry is not required. Significant background data movements are therefore allowed between the shared memory and the external circuitry, because means are provided for synchronizing and validating the data structures without high core overheads.

Preferably, steps b) and c) are repeated until said depth limit has been reached. Thereby, high amounts of data can be transferred by corresponding DMA transfers without interrupting the processor device. The DMA transfer may be triggered by hardware or software. The transaction log may be configurable or not.

The informing step may be performed by initiating an interrupt operation, e.g. triggering of an interrupt service routine at the processor device.

Furthermore, the transaction log may be transferred by an own DMA channel or may be appended to a data transfer. Alternatively, the transaction log my be retained in the connected circuitry, for interrogation following a qualifying interrupt.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following, the present invention will be described in greater detail based on a preferred embodiment with reference to the drawing figures, in which:

Fig. 1 shows a schematic block diagram of a digital signal processor connected to an ASIC; and

Fig. 2 shows flow diagrams of a data transfer method according to the preferred embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The preferred embodiment will now be described based on a pre-programmed DMA located on an ASIC-hosted interface to provide a data movement between a

shared memory 16 of a DSP or Microcontroller Unit (MCU) 10 and an ASIC 20, as shown in Fig. 1.

According to Fig. 1, the DSP 10 comprises a CPU 12 for controlling DSP operations based on a control program stored in a program memory (not shown). Furthermore, a host interface 14 is provided for controlling a data transfer to a slave interface 22 arranged at the ASIC 20. An interrupt control unit 17 is arranged in the DSP 10 to control the execution of interrupt service routines based on an interrupt control signal received from a resource manager 24 of the ASIC 20. Moreover, a slave interface 18 of the DSP 10 is controlled by a host interface 26 of the ASIC 20 so as to transfer data from the ASIC 20 to the shared memory 16 of the DSP 10.

In the ASIC 20, a memory 28 is provided for storing data to be transferred e.g. to the shared memory 16. The host interface 26 comprises a DMA controller for controlling DMA transfers via the slave interface 18 to the shared memory 16. Furthermore, a DMA transaction record or log comprising information about a DMA transfer history from the ASIC 20 to the DSP is stored at the host interface 26. The resource manager 24 is arranged to provide resource control of the ASIC 20 so as to enable the data transfer from the memory 28 to the host interface 26 under the control of the DMA controller.

As indicated by the dotted arrows in Fig. 1, a DMA transfer from the ASIC 20 to the DSP 10 comprises at least two phases, i.e. a first DMA transfer DMA #1 for transferring the respective data DATAX to the shared memory 16 of the DSP 10 and a directly or later following second DMA transfer DMA #2 for transferring the DMA transaction log from the host interface 26 to the shared memory 16. Thereby, high amounts of data can be transferred from the memory 28 of the ASIC 20 to the shared memory 16 of the DSP 10 by using DMA operations without involving the CPU 12. The CPU 12 is then informed of the data transfer by an interrupt control signal issued from the resource manager 24 to the interrupt control unit 17 after the transfer of the DMA transaction log.

An example for a data movement from the ASIC 20 to the DSP 10 is now described based on the flow diagrams given in Fig. 2.

In an initialization phase the DMA for the transfer of first data DATAX is set up in the ASIC 20 in step S101. Then, a DMA for the transfer of the DMA transaction log is also set up in the ASIC 20 in step S102. Thereby, both DMA transfer operations

required for the data movement from the ASIC 20 to the shared memory 16 of the DSP are initialized.

Then, a run-time DMA handling function is started at the DMA controller of the ASIC 20 to control the DMA transfer of the required amount of data. In step S201, the DMA transfer of the first data DATAX is triggered and an information indicating the transfer of the first data DATAX is added to the DMA transaction log (step S301). Similarly, subsequent DMA transfers of further data DATAY, DATAZ and DATAA are triggered in the subsequent steps S202 to S204, while the DMA transaction log is successively updated in steps S302 to S304. Thus, after the data transfer operation has been completed, the DMA transaction log contains transfer information specifying the data transferred to the shared memory 16 of the DSP 10.

When the DMA controller detects that the DMA transaction log has reached a predetermined depth limit, a transfer notification procedure is initiated by step S401. Then, a DMA transfer of the DMA transaction log is triggered in step S402 to transfer the DMA transaction log to the DSP 10 and store it in the shared memory 16. Then, the DMA controller triggers an interrupt to the CPU 12 of the DSP 10 by providing a corresponding control information to the resource manager 24 (step S403). In response to this interrupt, the CPU 12 qualifies or validates the data transferred to the shared memory based on the DMA transaction log also stored in the shared memory 16 (step S404). To achieve this, the interrupt control information supplied from the resource manager 24 to the interrupt control unit 17 may comprise a corresponding address information indicating the address of the DMA transaction log. The DMA transaction log may then comprise an information indicating the address ranges of the transferred data. The completion of the DMA movement in the transaction log may be indicated using DMA channel numbers

The proposed processor interfacing concept thus allows significant background data movement between the ASIC 20 and the shared memory 16 without introducing high core overheads at the DSP 10.

It is noted that the present invention is not restricted to the preferred embodiment described above, but can be used for any DMA transfer between processor devices and other circuitries connected to the processor device. Furthermore, the signaling of the DMA transaction log to the DSP 10 may be performed by any signaling option and is not restricted to an interrupt operation. The DMA transfer may

be performed to and from any memory-mapped location on the ASIC 20 or any other circuitry connected to the DSP 10. The transfer of the transaction log does not necessarily need its own DMA channel (e.g. DMA#2 in Fig. 1). It may be appended to a data transfer or may even by retained within the ASIC 20 for interrogation following a qualifying interrupt. This may include Interrupt Status. Thus, in Fig. 1, the DMA#2 is optional. The transaction log may be e.g. appended to the next data transfer. The DMA is not necessarily located on the host interface 26 of the ASIC 20, but may also hang on an ASIC OCB. Furthermore, the interrupt not necessarily has to be performed between two devices, but may also be triggered within the DSP 10. Thus, in Fig. 2, steps S102, S402 and S403 are optional steps for the specific case indicated in the configuration according to Fig. 1. The DMA transfer may be triggered also in software. The preferred embodiment may thus vary within the scope of the attached claims.

Claims

1. A method of performing a data transfer between a memory (16) of a processor device (10) and a circuitry (20) connected to said processor device (10), said method comprising the steps of:
   a) setting up a direct memory access (DMA) for transferring data at said circuitry (20);
   b) triggering a DMA transfer of said data to said processor device (10);
   c) adding said DMA transfer to a transaction log;
   d) providing said transaction log to said processor device (10), when said transaction log has reached a predetermined depth limit; and
   e) informing said processor device (10) of the availability of said transaction log.

2. A method according to claim 1, wherein said steps b) and c) are repeated until said depth limit has been reached.

3. A method according to claim 1 or 2, wherein said informing step is performed by initiating an interrupt operation.

4. A method according to claim 3, wherein said interrupt operation initiates an interrupt service routine.

5. A method according to any one of the preceding claims, further comprising the step of validating said transferred data at said processor device (10) based on said available transaction log.

6. A method according to any one of the preceding claims, wherein said circuitry is an ASIC.

7. A method according to any one of the preceding claims, further comprising the step of storing said transaction log in said memory (16).

8. A processor device having a memory (16) which can be accessed by a connected circuitry (20), said processor device (10) being arranged to validate data, transferred to said memory (16) by a direct memory access, based on a transaction log provided to said processor device.

9. A processor device according to claim 8, wherein said processor device (10) is arranged to validate said transferred data in response to an interrupt triggered by said connected circuitry (20).

10. A processor device according to claim 8 or 9, wherein said processor device is a digital signal processor (10).

11. An integrated circuit having means (22) for providing access to a processor device (10), said integrated circuit (20) being arranged to set up a direct memory access (DMA) for transferring data via said access means (22), to trigger a DMA transfer of said data, to add said DMA transfer to said transaction log, to provide said transaction log to said processor device when said transaction log has reached a predetermined depth limit, and to issue an information indicating the availability of said transaction log.

12. A integrated circuit according to claim 11, wherein said integrated circuit (20) is arranged to issue said information by triggering an interrupt.

13. An integrated circuit according to claim 11 or 12, wherein said integrated circuit is an ASIC.

14. A system for performing a data transfer between a memory (16) of a processor device (10) and a circuitry (20) connected to said processor device (10),

a) wherein said circuitry (20) is arranged to set up a direct memory access (DMA) for transferring data, to trigger a DMA transfer of said data to said processor device (10), to add said DMA transfer to a transaction log, to provide said transaction log to said processor device (10) when said transaction log has reached a predetermined depth limit, and to inform said processor device (10) of the availability of said transaction log; and

b) wherein said processor device (10) is arranged to validate said transferred data based on said provided transaction log.

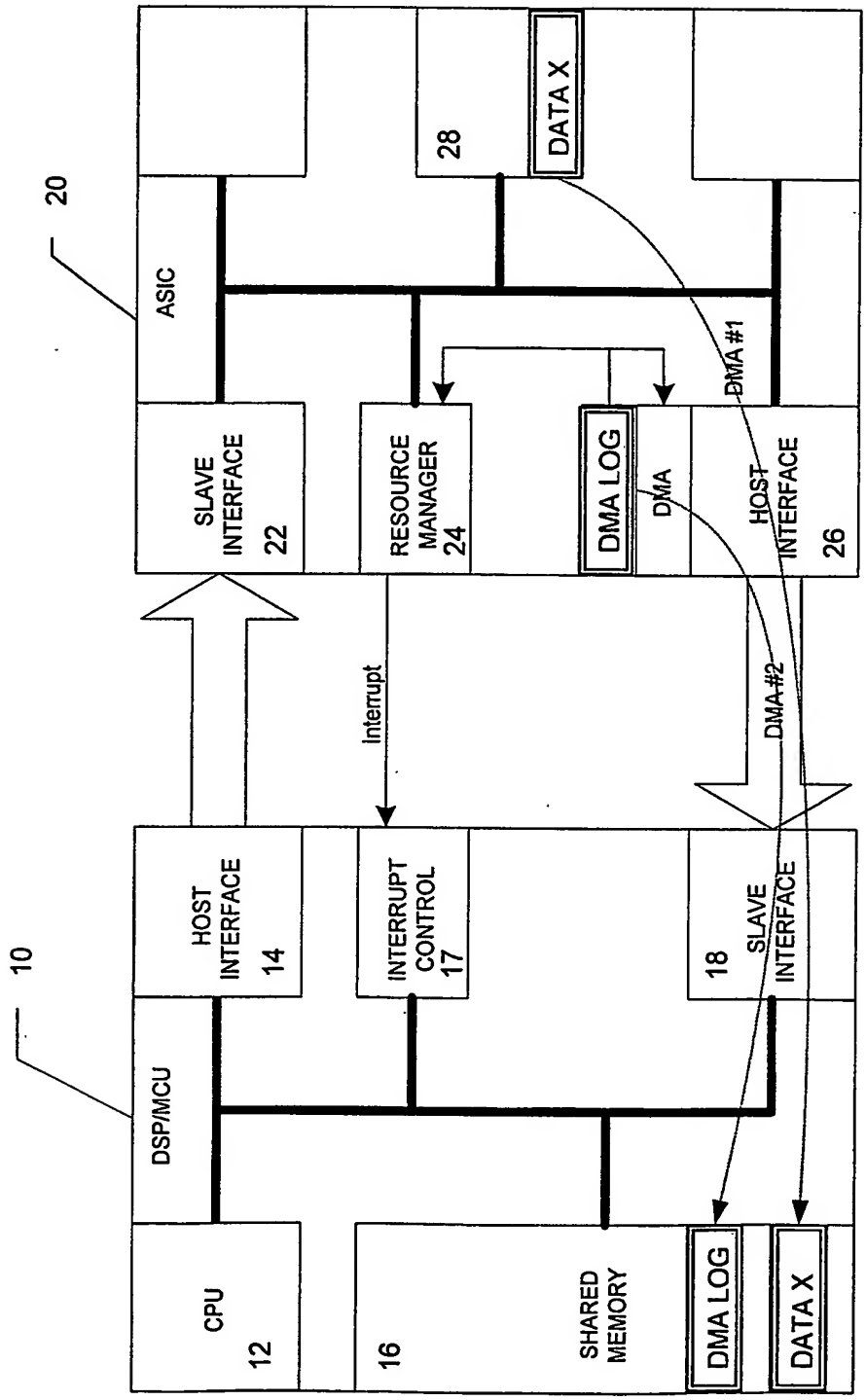

10
20
ASIC
28
DATA X
SLAVE INTERFACE 22
RESOURCE MANAGER 24
DMA LOG
DMA
HOST INTERFACE 26
DMA #1
DMA #2
Interrupt
HOST INTERFACE 14
INTERRUPT CONTROL 17
18 SLAVE INTERFACE
DSP/MCU
CPU 12
16
SHARED MEMORY
DMA LOG
DATA X

Fig. 1

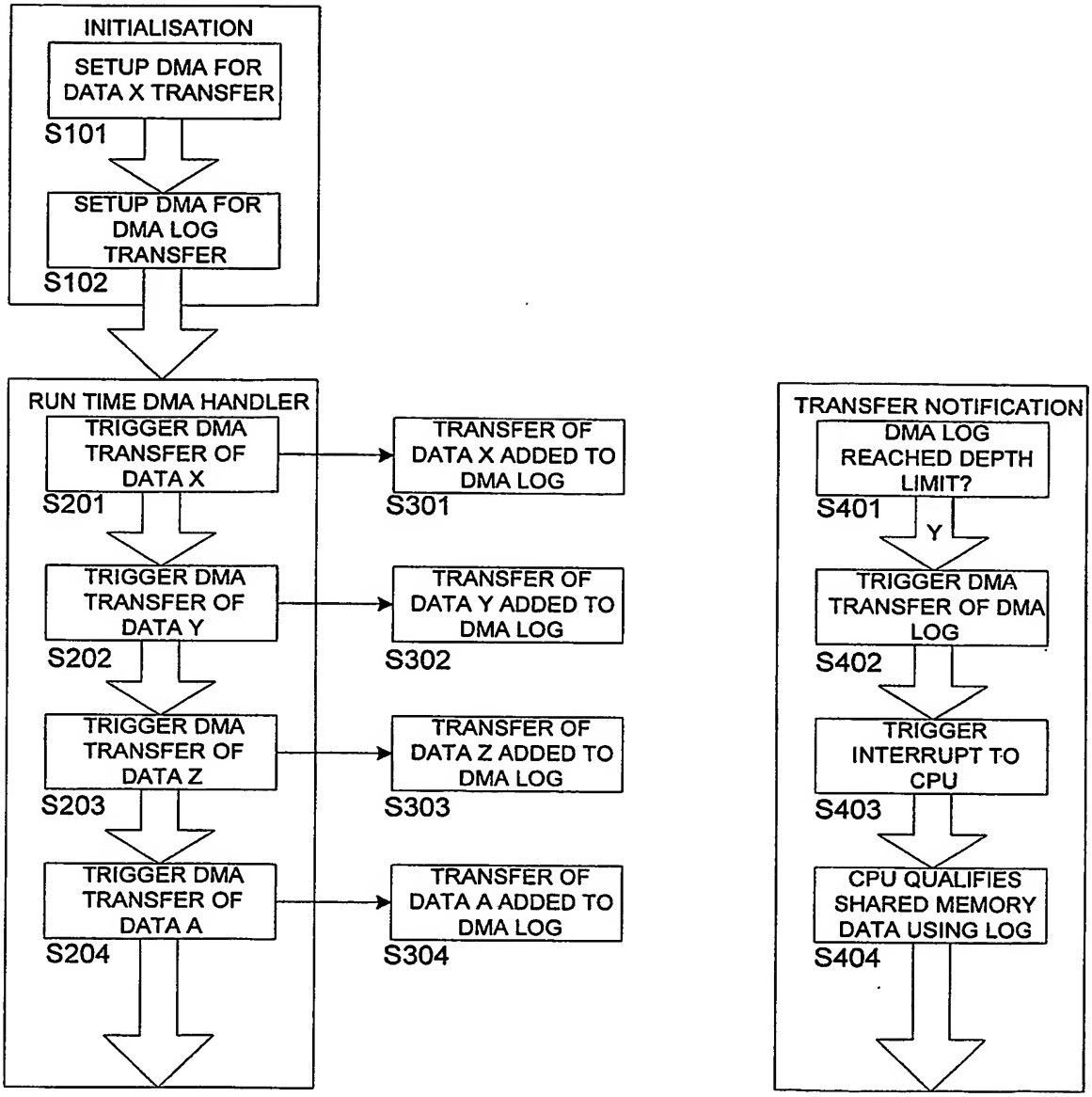

INITIALISATION
SETUP DMA FOR DATA X TRANSFER
S101
SETUP DMA FOR DMA LOG TRANSFER
S102
RUN TIME DMA HANDLER
TRIGGER DMA TRANSFER OF DATA X
S201
TRANSFER OF DATA X ADDED TO DMA LOG
S301
TRIGGER DMA TRANSFER OF DATA Y
S202
TRANSFER OF DATA Y ADDED TO DMA LOG
S302
TRIGGER DMA TRANSFER OF DATA Z
S203
TRANSFER OF DATA Z ADDED TO DMA LOG
S303
TRIGGER DMA TRANSFER OF DATA A
S204
TRANSFER OF DATA A ADDED TO DMA LOG
S304
TRANSFER NOTIFICATION
DMA LOG REACHED DEPTH LIMIT?
S401
Y
TRIGGER DMA TRANSFER OF DMA LOG
S402
TRIGGER INTERRUPT TO CPU
S403
CPU QUALIFIES SHARED MEMORY DATA USING LOG
S404

Fig. 2

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP 01/15347

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F13/28

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4 805 137 A (GRANT JOHN L ET AL) 14 February 1989 (1989-02-14) column 1, line 62 - line 66 column 3, line 2 - line 14; figure 1 ---	1-14
A	EP 0 908 830 A (IBM) 14 April 1999 (1999-04-14) abstract ---	1-14
A	US 4 959 782 A (TULPULE BHALCHANDRA R ET AL) 25 September 1990 (1990-09-25) column 3, line 50 -column 4, line 18; figure 1 -----	1-14

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 February 2002	04/03/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Juenger, B

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP 01/15347

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 4805137 A	14-02-1989	NONE	
EP 0908830 A	14-04-1999	US 5968158 A	19-10-1999
		EP 0908830 A1	14-04-1999
		JP 11175491 A	02-07-1999
		US 6233643 B1	15-05-2001
US 4959782 A	25-09-1990	US 5093910 A	03-03-1992